Assignment 4: Linear Algebra library – Armadillo

June 1, 2017

1 Background

In this assignment, we will introduce you to the Armadillo linear algebra library. This last assignment will serve as a recap through the course and is a test for you to see whether you are able to write programs with C++ on your own. In contrast to previous assignments, we are not going to present you with a header file with functions to implement, instead you are asked to plan and write your program freely.

Your task is to use Armadilllo to implement two well known statistical algorithms for classification: Nearest Neighbours and Logistic Regression. Both algorithms assign a label to an unseen point. For example, if the point is an image of a lung patient, the algorithms will decide whether the patient is suffering from an illness or not. For simplicity, we will only require binary classification, i.e. the labels to be assigned are either 1 or -1.

2 Exercise 1: Nearest neighbour classification

The nearest neighbour algorithm is a simple algorithm that assigns a label using a majority voting scheme. A previously unseen point is compared to a set of points with known labels. The algorithm picks the k points with closest distance and assigns the label of the majority of points. More formally, given a set of points $\{(x_1, y_i), \ldots, (x_N, y_N)\}$ and a number of neighbours, k we assign a label y to a new point x as

- Compute distances $d_i = \|x_i - x\|$
- Create a set of label-distance pairs P with $P_i = (y_i, d_i)$, $i = 1, \ldots, N$.
- Sort P ascending by d_i(i.e. afterwards P_1 is the pair with smallest d_i)
- Assign to y the label which appears most often among $P_1, \ldots, P_k$.

Your program implemented in the source file "NearestNeighbours.cpp" has to do the following:

- Read "dataX.dat", "dataXtest.dat" and "dataY.dat" from disk. "dataX.dat" contains the points x_i in its rows. They are stored as numbers separated by a white space (i.e. tab or space). "dataY.dat" contains the label y_i, one in each row. You can assume that the only labels appearing are 1 or -1. "dataXtest.dat" contains test points.
- For each point in "dataXtest.dat", assign the apropriate label using the nearest neighbour algorithm with $k = 5$.
- Write a new file "NN.dat" containing the assigned labels in the same format as "dataY.dat".

3 Exercise 2: Logistic Regression

Logistic regression assigns the label y to a point x using a previously chosen linear function $f(x) = w^T x$. To find the label, we use $y = \text{sign}(f(x))$ (i.e. if f(x) is positive, the label is 1, else the label is -1). We find w by minimizing the function

$$L(w) = \frac{1}{N} \sum_{i=1}^{N} \log(1 + \exp(-y_i w^T x_i))$$

For this assignment, we use an algorithms called gradient descent to find the optimal w. This algorithms uses the gradient of $L(w)$:

$$\frac{\partial}{\partial w} L(w) = -\frac{1}{N} \sum_{i=1}^{N} y_i \frac{1}{1 + \exp(y_i w^T x_i)} x_i$$

For an existing initial guess of w (for example $w = 0$), we can find a better estimate using

$$w \leftarrow w - \alpha \cdot \frac{\partial}{\partial w} L(w) \ ,$$

where $0 < \alpha < 1$ is a chosen learning rate. We can repeat this, until our solution is close enough to the optimal solution. We consider our choice of w to be close enough when the above update role does not change w very much any more, i.e. when $\| \frac{\partial}{\partial w} L(w) \| < \epsilon$, for some chosen tolerance ϵ.

Your program implemented in the source file "LogisticRegression.cpp" has to do the following:

- Read "dataX.dat", "dataXtest.dat" and "dataY.dat" from disk (same as Exercise 1).
- Find the optimal w using the rules above. The final solution should obey $\epsilon = 10^{-7}$
- For each point in "dataXtest.dat", assign the apropriate label using the assignment rule $y = \text{sign}(f(x))$.
- Write a new file "LogReg.dat" containing the assigned labels in the same format as "dataY.dat".

Hints:

- The documentation of Armadillo is at `http://arma.sourceforge.net/docs.html`.
- It might be helpful to implement file input and output in a common file.
- The version of Armadillo provided is configured to be stand-alone, which means that some advanced operations: solving systems of equations or eigenvalues are not available. However this should not be relevant for the assignment.
- A test for whether you implemented the algorithms correctly is to copy the points from "dataX.dat" into "dataXtest.dat". In this case, you should expect that almost all of the labels assigned by the algorithm should be the same as in "dataY.dat".